IN THE CLAIMS

Please delete claims 10-13.

Claims 1-9 and 14-24, which have been amended, are shown below. Attached is Exhibit C showing the amendments made to the claims. Please also add claims 26-47 as shown below.

1.   (Amended)   A system for storing, using and protecting access to sensitive data, comprising:

non-volatile storage;

a hidden storage location;

a first key derived from said sensitive data;

means to prevent use of the first key by programs running in the normal operating mode of the system; and,

means to allow use of the first key by a program running in a restricted operating mode of the system.

2.   (Amended)   The system recited in Claim 1, wherein the sensitive data is the first key.

3.   (Amended)   The system recited in Claim 1, wherein the sensitive data is derived using an application of a one-way cryptographic digest function of the first key.

4.   (Amended)   The system recited in Claim 3, wherein the sensitive data is a second key retrieved from encrypted data stored in a storage medium, where the encrypted data is encrypted with a key derived from the first key.

5.   (Amended)   The system recited in Claim 1, wherein firmware in a ROM or flash ROM controls the system during the system initialization process that begins in response to a power-on or reset signal.

6.   (Amended)   The system recited in Claim 1, wherein:

the non-volatile storage is non-volatile random access memory with read and write access controlled by a latch;

the latch is opened at the start of the system initialization process due to a hardware function responding to a power-on or reset event, thereby enabling system access to the non-volatile random access memory; and

the latch is closed during the system initialization process, thereby denying system access to the non-volatile random access memory until the next start of system initialization.

7.   (Amended)   The system recited in Claim 1, wherein:

the hidden storage location is in a system management random access memory which is not accessible by any program running in the normal operating mode of the system; and

the restricted operating mode is a system management mode in which access to the system management random access memory is permitted.

8.   (Amended)   The system recited in Claim 1, wherein:

the hidden storage is restricted for access by the operating system only, and is not accessible

by any application program that runs in the normal operating mode of the system; and

the restricted operating mode is controlled by a CPU protection ring reserved for use by

operating system software.

A1
Cancld -

9.   (Amended)   A system for hiding a cryptographic key in storage, comprising

power-on software that:

reads a key from non-volatile storage;

closes access to the non-volatile storage such that access does not become available again

until the next system reset;

writes sensitive data derived from the cryptographic key to a hidden address space; and,

wherein only a program that runs in a restricted operational mode of the system has

access to the sensitive data in the hidden address space.

A2
Cm't

14.   (Amended)   A method of controlling access to data by an application program by

restricting the use of a cryptographic key for the application program on a device, comprising:

providing a first key known to a cryptographic processing module;

providing an application container data structure that contains a cryptographically sealed

form of the data for the application program to access;

performing a cryptographic gatekeeping function that computes a cryptographic digest of a

portion of an in-memory image of the application program to compute a cryptographic digest of the

application; and

performing an integrity-check by the cryptographic processing module by examining the application container data structure, the cryptographic digest, and the first key to determine if the application program is allowed to unseal the cryptographically sealed form of the data.

15. (Amended) The method recited in Claim 14 further comprising performing a privacy operation by the cryptographic processing module that encrypts or decrypts the cryptographically sealed form of the data in the application container data structure using a key derived from at least the first key and cryptographic digest, and when the cryptographically sealed form of the data is to be encrypted, the crytographic processing module adds to the application container data structure the cryptographic digest before the encryption is performed.

16. (Amended) The method recited in Claim 14 further comprising providing an authorization buffer that specifies the result of the integrity-check, and wherein the cryptographic gatekeeping function confirms that the application program is allowed to unseal the cryptographically sealed form of the data.

17. (Amended) The method recited in Claims 14 wherein the integrity-check includes:

deriving a cryptographic variable from the cryptographic digest and the first key; and

using the cryptographic variable to check a message authentication code that is stored in the application container data structure.

18. (Amended) The method recited in Claims 14 wherein the integrity-check includes:

decrypting data derived from the application container data structure using a key derived from the first key to create a resulting value and comparing the resulting value to data derived from the cryptographic digest; and

allowing access to the cryptographically sealed form of the data if the resulting value is the same as the data derived from the cryptographic digest.

19.   (Amended)   The method recited in Claims 15 wherein the privacy operation includes:

deriving a cryptographic variable from the cryptographic digest and the first key, wherein the cryptographic variable is used to decrypt or encrypt a portion of the application container data structure.

20.   (Amended)   The method recited in Claim 19 wherein the cryptographic variable is derived with one or more applications of a hash function by concatenating dependant values in a particular order.

21.   (Amended)   The method recited in Claim 14 wherein a portion of the cryptographic processing module executes during an system management interrupt.

22.   (Amended)   A method for authenticating an identified application program on an identified device to another computing machine comprising an authentication server with the help of another computing machine comprising a device authority, the method comprising:

performing a first cryptographic enrollment operation during a system management interruption on the identified device producing a result that is sent to the device authority;

performing a second cryptographic enrollment operation during the system management interruption on the identified device processing a value generated by the device authority that is received by the identified device;

performing a first cryptographic registration operation during the system management interruption on the identified device producing a result that is sent to the authentication server;

performing a second cryptographic registration operation by the authentication server producing a cryptographic variable that is stored for use during the authentication method;

performing a first cryptographic authentication operation during the system management interruption on the identified device producing authentication data that is sent to the authentication server; and

performing a second authentication cryptographic operation by the authentication server on the authentication data received from the identified device using at least the cryptographic variable to determine the result of the authentication.

23.   (Amended)   A method for authenticating an identified application program on an identified device, or for providing a second factor for identifying a user of the identified device to another computing machine comprising an authentication server, the method comprising:

performing an enrollment process including communicating with a device authority and an authentication server to create an application container data structure on the device, wherein the application container data structure is cryptographically associated with the application program;

storing credential information, wherein the authentication server stores a cryptographic variable for the application container data structure;

unsealing the application container data structure that stores the credentials;

modifying the credentials;

resealing the application container data structure, wherein at least part of said resealing occurs during an SMI on the same CPU that executes the code of the application program;

sending identifying information and data derived from the resealed application container data structure to the authentication server;

receiving the identifying information and the data derived from the application container data structure;

using the identifying information to lookup or compute a cryptographic variable to unseal the application container data structure;

authenticating the identified application program and the identified device if the unsealed application container includes acceptable values; and

storing a key associated with the application container data structure.

A2
Cm't

24.   (Amended)   A system for creating and utilizing one or more virtual tokens on a device for the purpose of at least one of authentication, privacy, integrity, authorization, auditing, and digital rights management, each of said one or more virtual tokens having a corresponding type, the system comprising:

an application program for each of said corresponding type of virtual tokens;

an application container for each of said corresponding type of virtual tokens; and

a cryptographic gatekeeping component that computes a cryptographic digest of a calling application that is requesting cryptographic services of a cryptographic processing component,

wherein the cryptographic processing component is accessed via the cryptographic gatekeeping component,

wherein the cryptographic processing component knows a first key and a public key,

wherein the cryptographic processing component performs cryptographic sealing and unsealing of application container data structures, where a portion of the cryptographic operations are performed during a system management interrupt,

wherein the cryptographic processing component checks the integrity of the calling application by checking a digital signature of a portion of the calling application's code or static data, using the public key known to the cryptographic processing component and a cryptographic digest value,

wherein the cryptographic digest value includes a recently computed cryptographic hash of a portion of the calling application's in-memory image, and

wherein the cryptographic gatekeeping and cryptographic processing component

A2 *cnceld.*

a) derive a key for unsealing the application container data structure from the first key and cryptographic digest,

b) use the derived key to check the message authentication code on the application container data structure, and returns an error if the message authentication code is correct, and

c) use the derived key to decrypt the data in the application container data structure and return it to the application.

25.   (Amended)   A method of securely associating a private key with an application program associated with a device, comprising:

creating an application container that contains private keys secured by a key associated with the application program and the device.

Please add the following claims:

26.   (New) A system of restricting access to sealed data to an application program on a computer comprising:

a first key in hidden storage on the computer;

a cryptographic gatekeeper module that runs in a restricted mode and computes a cryptographic digest of a portion of the application program;

an application container data structure that contains the sealed data; and

a means to determine whether to grant the application program access to the sealed data using the first key and the cryptographic digest.

27.   (New) The system of Claim 26, wherein the application program is part of an operating system kernel.

28.   (New) The system of Claim 26, wherein data derived from the application container data structure is validated using the first key before granting access to the sealed data.

29.   (New) The system of Claim 26, further comprising a cryptographic processing module that conducts a privacy operation by encrypting or decrypting the sealed data using a key derived from the first key and the cryptographic digest.

30.   (New) The system of Claim 29, wherein the privacy operation, prior to encrypting or decrypting the sealed data, adds the cryptographic digest to the application container data structure.

31.   (New) The system of Claim 26, further comprising a cryptographic processing module that performs a tamper detection operation by computing a message authentication code based on a key derived from the first key and the cryptographic digest.

32.   (New) The method of Claim 26, wherein the cryptographic key is derived from a plurality of data items chosen from the group consisting of:

the first key;

the cryptographic digest;

a password; and,

values passed to the cryptographic gatekeeping function.

33.   (New) The method of Claim 32, wherein the cryptographic key is derived using a hash function.

34.   (New) An enhanced computing device, comprised of:

a processor to execute a plurality of application programs in a normal mode;

a security kernel that executes in a restricted mode;

a key that is accessible by the security kernel when said processor is executing in the restricted mode, where the security kernel uses the key to authenticate an application program on the computing device and provides cryptographically secure data for use by the application program.

35.   (New) The enhanced computing device of Claim 34, wherein the key is associated with the device.

36. (New) The enhanced computing device of Claim 35, wherein the cryptographically secure data will not be accessible when moved to a different device.

37. (New) The enhanced computing device of Claim 34, wherein the container will not function properly when moved to a different device.

38. (New) The enhanced computing device of Claim 34, where restricted mode is SMM.

39. (New) The system of Claim 1, wherein the first key is a shared key for use in a symmetric key cryptosystem.

40. (New) The system of Claim 1, wherein the first key is a private key for use in a public key cryptosystem.

41. (New) The system of Claim 1, wherein the normal operating mode is one of a kernel mode and a user mode in a 32-bit operating system environment.

42. (New) The system of Claim 1, wherein the restricted operating mode is system management mode.

43. (New) A method to assess the integrity of an executable in-memory image of a program, wherein said method examines a portion of the bytes of the program and uses the result of the integrity assessment to restrict access of the program to cryptographic services or data.

44. (New) The method of claim 14 where restricted mode is the kernel mode of a 32-bit operating system.

45. (New) The method of Claim 14, wherein the cryptographic key is derived from a plurality of data items chosen from the group consisting of:

the first key;

the cryptographic digest;

a password; and,

values passed to the cryptographic gatekeeping function.

46. (New) The enhanced computing device of Claim 34, wherein the container will not function properly when moved to a different device.

47. (New) The enhanced computing device of Claim 34, where restricted mode is SMM.